



Licenciatura Engenharia Informática e Multimédia

Modelação e Programação – MoP

Relatório Trabalho Prático 4 Parte B

Docente André Gomes

Trabalho realizado por:

Fábio Dias, nº 42921

Índice

1. Descrição do Projeto	4
2. Descrição das Classes.....	5
3. Ficheiro de Gravação	7
4. Notas.....	9

Índice de Figuras

Figura 1 - UML.....	6
Figura 2 - Ficheiro de Gravação XML	8

1. Descrição do Projeto

Para este trabalho prático decidi fazer um jogo onde existe uma personagem que possui três atributos e seis vagas para equipamento. A personagem possuirá uma arma básica no início do jogo e, à medida que se aventura, derrotará inimigos com características geradas aleatoriamente e, conseqüentemente, receberá novos equipamentos.

O jogador também tem de gerir o seu inventário pois tem vagas limitadas. Desta forma, o jogador decide se quer guardar itens, equipá-los ou desequipá-los ou descartá-los do inventário.

O objetivo do jogo é o jogador chegar o mais longe possível até, eventualmente, perder todos os seus pontos de vida. Assim, volta ao menu onde pode mudar os seus itens por itens melhores que arranhou nas suas aventuras, melhorando os seus atributos, e voltar a aventurar-se.

Dado que podemos ter várias personagens e podemos continuar o progresso anterior, é necessário um sistema de gravação. Este será feito por um documento XML que guarda o nome da personagem, os itens equipados e os itens no inventário.

De forma a realizar a parte gráfica do programa, usado a biblioteca Swing do Java. Definimos uma *frame* que possuirá tudo da aplicação e, partindo do uso de *panels* e *labels*, estabelecemos quatro menus importantes: o *Opening Menu*, o *Main Menu*, o *Armory Menu* e o *Adventure Menu*.

O *Opening Menu* possui as decisões de começar um novo jogo ou carregar um jogo previamente guardado.

No *Main Menu* existe a possibilidade de partir para a aventura, sala de equipamentos, guardar o jogo e sair da aplicação.

No *Armory Menu* é onde podemos mudar o equipamento do nosso jogador e podemos começar a aventura ou voltar ao *Main Menu*.

Finalmente, no *Adventure Menu* é onde podemos combater com os inimigos e ganhar melhores equipamentos.

2. Descrição das Classes

Dado que tanto a nossa personagem como os nossos inimigos partilham atributos e os métodos correspondentes, foi criado uma classe abstrata “GameCharacter” e duas classes estendem desta: “Player” e “Enemy”. De forma a adicionar alguma variedade aos inimigos, foi declarada uma *String* que possuirá a sua raça que pode variar entre diversos tipos de criaturas míticas. Foram também criados dois arrays de *String*’s, um possui um conjunto de nomes que os inimigos podem ter e outro as raças. Quanto à classe Player, esta possuirá dois array de *Items*, um que será o inventário e outro que são os equipados. Foram criados diversos métodos de forma a gerir ambos os arrays.

Como já mencionado, também foi criada uma classe *Item* que servirá para qualquer tipo de item que o jogador pode possuir. Possui atributos, dependendo da sua raridade e que tipo de item é.

Foi desenvolvido uma classe *Item Generator* que, como o nome indica, vai gerar um novo item sempre que o jogador derrota um inimigo. O número de inimigos derrotados até ao momento terá influência quanto à raridade do objecto. Quanto mais raro, melhor serão os seus atributos.

Seguindo a mesma lógica, foi implementada a classe *EnemyGenerator* de forma a criar um inimigo aleatório e instanciada na classe “Game”.

Também mencionado o sistema de gravação, foi criado uma classe estática *Save System* que cria o ficheiro, caso este não exista, grava o progresso actual, assim como carrega o progresso guardado previamente noutra personagem.

Por fim, temos o aclamado “*game loop*” onde existe o método *main* e onde ocorrem todas as interações com o jogo. Vamos ter diversos menus e, para identificar em qual nos encontramos, recorri a um *enum GameState*.

Dentro da classe “Game” também foi implementada o menu gráfico. Desta forma, temos novos métodos como o que inicializa a *frame* e inicializa os menus todos mencionados acima. Quanto a outros métodos são específicos à batalha entre o jogador e o inimigo, assim como a atualização da informação do jogador e inimigo.

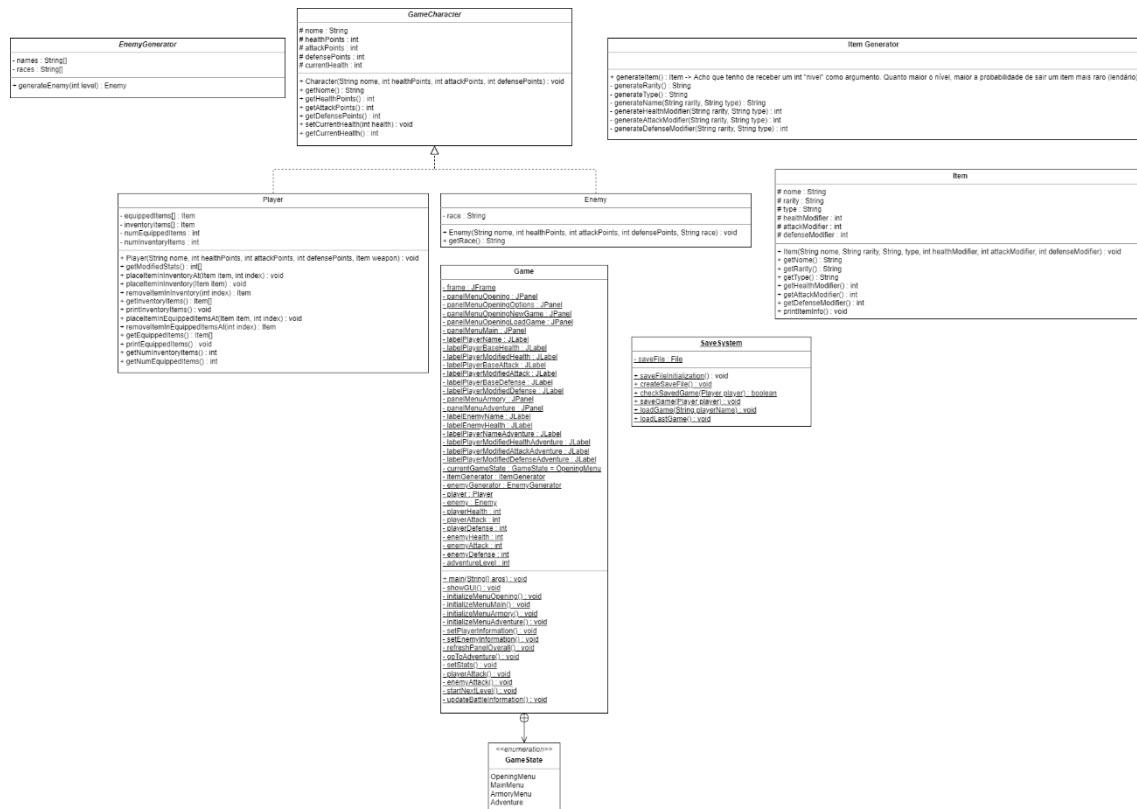


Figura 1 - UML

3. Ficheiro de Gravação

Para o ficheiro de gravação, foi necessária a criação de um documento de XML. A organização do mesmo é a partir de um elemento base “SaveData” que possui dois elementos “SaveGames” e “LastSavedGame”. Esta primeira guarda o progresso do jogo a partir da criação de um elemento “Game”, que guarda o nome da personagem no elemento “Name”, e os itens que o jogador possui pelo elemento “Items”. Este é dividido em dois: “Equipped” que cria seis elementos filho que correspondem às vagas dos itens que estão equipados actualmente e outro “Inventory” que tem todos os itens guardados no inventário.

O elemento LastSavedGame possui o nome da personagem cujo progresso foi guardado da última vez como referência.

```

1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2  <SaveData>
3      <SaveGames>
4          <Game>
5              <Name>Teste</Name>
6              <Items>
7                  <Equipped>
8                      <Weapon>
9                          <Name>Basic Sword of the Basics</Name>
10                         <Rarity>Common</Rarity>
11                         <Health>0</Health>
12                         <Attack>20</Attack>
13                         <Defense>0</Defense>
14                     </Weapon>
15                     <Helmet/>
16                     <Armory/>
17                     <Pants/>
18                     <Gloves/>
19                     <Boots/>
20                 </Equipped>
21                 <Inventory>
22                     <Item>
23                         <Name>Pants Equipamento</Name>
24                         <Rarity>Uncommon</Rarity>
25                         <Type>Pants</Type>
26                         <Health>71</Health>
27                         <Attack>0</Attack>
28                         <Defense>33</Defense>
29                     </Item>
30                 </Inventory>
31             </Items>

```

Figura 2 - Ficheiro de Gravação XML

4. Notas

O ItemGenerator ainda não implementa nomes variados.

Também não foi implementado os métodos de Load Game e Load Last Game do Save System.

Para este jogo foram pensadas diversas mecânicas que tornariam o jogo mais dinâmico. Uma delas era termos diversos ataques que poderiam ser desbloqueados por certos tipos de equipamento. Neste caso atual, temos apenas um ataque mas, com esta mecânica, conseguiríamos ter até quatro ataques distintos pois os itens poderiam fornecer aos jogador esses mesmos.

Outra ideia pensada foi termos efeitos. Caso estes diversos ataques existissem, seria possível termos efeitos que favoreciam o jogador e desfavoreciam o inimigo, assim como o inimigo podia ter efeitos que o favoreciam e que desfavoreciam o jogador. Estes podiam ser “Stun” onde a personagem perderia o turno, “Burn” onde a personagem perdia uma pequena porção de vida durante alguns turnos. “Morale Boost” que aumentaria o dano e a resistência da personagem.